Indistinguishability between quantum randomness and pseudorandomness under efficiently calculable randomness measures

Toyohiro Tsurumaru,[1,*] Tsubasa Ichikawa,[2,†] Yosuke Takubo,[3,4,‡] Toshihiko Sasaki,[5,§] Jaeha Lee,[6,‖] and Izumi Tsutsui[3,¶]

[1]*Mitsubishi Electric Corporation, Information Technology R&D Center, Kanagawa, 247-8501, Japan*
[2]*Center for Quantum Information and Quantum Biology (QIQB), Osaka University, Osaka 560-0043, Japan*
[3]*Institute of Particle and Nuclear Studies, High Energy Accelerator Research Organization (KEK), Ibaraki 305-0801, Japan*
[4]*Graduate University for Advanced Studies (SOKENDAI), Hayama 240-0193, Japan*
[5]*Photon Science Center, Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan*
[6]*Institute of Industrial Science, The University of Tokyo, Chiba 277-8574, Japan*

(Received 5 October 2023; accepted 17 January 2024; published 28 February 2024)

We present a no-go theorem for the distinguishability between quantum random numbers (i.e., random numbers generated quantum mechanically) and pseudorandom numbers (i.e., random numbers generated algorithmically). The theorem states that one cannot distinguish these two types of random numbers if the quantum random numbers are efficiently classically simulatable and the randomness measure used for the distinction is efficiently computable. We derive this theorem using the properties of cryptographic pseudorandom number generators, which are believed to exist in the field of cryptography. Our theorem is found to be consistent with the analyses on the actual data of quantum random numbers generated by IBM Quantum and also those obtained in the Innsbruck experiment for the Bell test, where the degrees of randomness of these two set of quantum random numbers turn out to be essentially indistinguishable from those of the corresponding pseudorandom numbers. Previous observations on the algorithmic randomness of quantum random numbers are also discussed and reinterpreted in terms of our theorems and data analyses.

DOI: 10.1103/PhysRevA.109.022243

I. INTRODUCTION

In quantum mechanics, it is widely believed that transitions between two states occur randomly, and that this randomness is genuine in the sense that it does not admit causal descriptions by means of determinants called hidden variables. This belief has been supported by numerous experiments (Bell tests) [1–14] to examine the Bell inequality [15] which ended up in excluding any local hidden variable theories. Besides, no experiments so far signal even nonlocal hidden variable theories including Bohm's one [16,17], suggesting the implausibility of causal descriptions underlying the quantum randomness.

Meanwhile, the genuineness of quantum randomness has become one of the bases for applications of quantum physics, such as in quantum cryptography [18] and random number generation [19]. It is also considered crucial in quantum mechanics to uphold the no-cloning theorem and ensure the consistency between quantum nonlocality and special relativity [20].

In view of the firmly established foundational trait and the ongoing successful applications, it is somewhat surprising that we still have little understanding on how different the quantum randomness is with respect to nonquantum ones. Recently, we have seen several attempts [21–25] designed to characterize the quantum randomness quantitatively using some randomness measures based on the theory of algorithmic complexity [26] and/or the statistical homogeneity of distribution. The algorithmic complexity quantifies the upper bounds of the memory size to generate the bit string with a universal Turing machine, whereas the statistical homogeneity evaluates relative frequency of the binary values of the bit string.

Among the randomness measures mentioned in these attempts, the Lempel-Ziv (LZ) complexity and the Borel normality appear particularly handy and commonly used. Here, the LZ complexity [27], which has been utilized to evaluate the complexity of nonlinear dynamical systems in physics [28,29], estimates the difficulty of compressing the given bit string with the LZ76 algorithm [27]. On the other hand, the Borel normality [30], which has been employed to compare the randomness of quantum systems with those of the other systems [21–25], quantifies the difference between the distribution of substrings in the given bit string and the uniform distribution thereof.

With these two measures, previous data analyses led us to two intriguing observations:

(i) The bit strings generated from a quantum random number generator (QRNG) have statistically larger values of the

[*] tsurumaru.toyohiro@da.mitsubishielectric.co.jp
[†] ichikawa.tsubasa.qiqb@osaka-u.ac.jp
[‡] yosuke.takubo@kek.jp
[§] sasaki@qi.t.u-tokyo.ac.jp
[‖] lee@iis.u-tokyo.ac.jp
[¶] izumi.tsutsui@kek.jp

Borel normality, which implies that the generated bit strings are less random than those generated from nonquantum pseudorandom number generators (PRNGs) such as the Mersenne twister [25], despite the fact that by definition the latter is actually less random than the former (see Sec. II A 3 for detail). This has been argued to be caused by the effects of unknown biases coming from experimental imperfections for the implementation of the QRNG.

(ii) The bit strings generated from the Bell tests (another type of QRNG) in [4,31] appear to have smaller values of the LZ complexity when the corresponding Bell tests violate the Bell inequality more strongly [24], which again implies that the generated bit strings are less random when they are generated nonlocally. In other words, the degree of algorithmic randomness appears to be anticorrelates with the degree of nonlocality.

The above observations are clearly in conflict with our belief on the genuineness of quantum randomness and cause us to doubt the validity of the randomness measure employed, or even to question the implicit assumption on the very existence of measures capable of distinguishing QRNGs from PRNGs, possibly under the influence of experimental imperfections.

In fact, it is not difficult to infer that the above assumption may be untenable. For example, suppose that one is given equal-weight superposition states of quantum two-level systems (qubits). By repeating ideal projection measurements, one will obtain a series of genuine random outcomes of binary numbers, and, hence, a qubit may be regarded as the "true random number generator" (TRNG) (our more technical definition of the TRNG is given in Sec. II). On the other hand, PRNGs for commercial use pass the randomness tests such as NIST suites, and are believed to imitate the TRNG with reasonable precisions, as far as the bit strings are finite and randomness tests are implementable in finite durations. This seems to contradict the fact that by definition PRNGs are less random than the TRNG. This indicates that, in practice, PRNGs will be able to mimic QRNGs at least for an ideal case.

On the basis of the above consideration, in this paper we present a no-go theorem on the above assumption. We derive this theorem by using the properties of cryptographic PRNGs (CPRNGs; see, e.g., Refs. [32,33]), which are believed to exist in the field of cryptography. To be more explicit, we shall show that, if there exists a CPRNG, whose outcomes are indistinguishable from those of the TRNG (under any polynomial time algorithm within a negligible error margin), then any efficiently computable randomness measures cannot distinguish the outcomes of systems which are "efficiently classically simulatable" [34] (which do not necessarily behave as the TRNG) from those of the PRNGs with a negligible error margin. Since some quantum systems are known to be efficiently classically simulatable, our no-go theorem is applicable to those systems. We also provide examples where the counterintuitive observations (i) and (ii) can be understood more easily as a consequence of the PRNGs capable of mimicking the QRNGs. This illustrates that the observations (i) and (ii) are actually consistent with our theorem.

In concrete terms, for observation (i) we find that, if the length of the bit string and the relative frequency of bit values are comparable between the data set generated by IBM Quantum and those generated by PRNGs, then the distributions of the LZ complexity and Borel normality are almost identical between IBMQ and PRNGs. The difference between them reported in [25] can be attributed to the fact that the relative frequencies were not aligned properly.

For observation (ii), we first point out flaws in the data analyses in [24] (see the last two paragraphs of Sec. III B 1 a). Motivated by this, we reanalyze the data set obtained in the Innsbruck experiment for the Bell test [31] and compare the results with those obtained from PRNGs. As with the quantum coin tosses, this analysis also shows that the distributions of the LZ complexity and Borel normality are comparable in the actual data set and PRNGs. In addition, we reject the hypothesis of the existence of an anticorrelation between the degree of randomness and that of the violation of the Bell inequality. We shall also exhibit a correlation between the LZ complexity and the binary entropy of the relative frequency, which indicates that systematic errors latent in the data acquisition process influence the degree of randomness.

This paper is organized as follows: In Sec. II we introduce notions relevant to our theoretical analyses and present our main results including the no-go theorem. In Sec. III we make the empirical analyses mentioned above to support our results and argue their implications and interpretations with respect to preceding works. Section IV is devoted to our conclusion.

II. OUR NOTION OF RANDOMNESS AND THE MAIN RESULTS

As announced in Introduction, we show the indistinguishability of the outcomes of the QRNGs and PRNGs with the use of any efficiently calculable randomness measures. Proof of this assertion requires several theoretical notions originating from cryptography, but possibly unusual in quantum information science. We hereafter make a review of these important notions (with slight generalizations if necessary), rigorously state our assertion, and give proof thereof.

We begin by summarizing the terminology. A quantum random number generator (QRNG) is literally a quantum system that outputs a random value. A pseudorandom number generator (PRNG), on the other hand, is a deterministic computation algorithm which transforms a short random bit sequence into a seemingly random long bit sequence (see Sec. II A 3 for details).

References [27,30] discuss indices I which can be computed from a random bit sequence X and measure the randomness of X. Hereafter such an index $I(X)$ will also be called a *randomness measure*. The references above also suggest the possibility that certain types of $I(X)$, such as the LZ complexity (see Sec. III B 1 a) and the Borel normality (see Sec. III B 1 b), can be used to distinguish PRNGs and QRNGs.

Our main message in this paper is that such an index I is not in fact feasible, in light of the knowledge of modern cryptography. The basic flow of the discussion is as follows.

While Refs. [27,30] placed no particular constraints on I, in the real world, one needs at least the following two conditions, for I to be able to distinguish PRNGs and QRNGs,

(1) There is a function $D_I : I \to \{0, 1\}$ that maps index I to one bit. The distribution of $D_I(I(X))$ differs depending

on whether the random number X is output from QRNG or PRNG.

(2) Indices I, D_I can be calculated in a realistic time.

(See Sec. II B, Definitions 3 and 4 for details.) However, if any randomness measure I can satisfy these conditions, it immediately leads to a contradiction with the common and widely used assumption of cryptography, namely, the existence of cryptographic PRNGs. In other words, as long as we accept the common assumption of cryptography, we can show that there is no good randomness measure I that can distinguish between QRNG and PRNG.

The meanings of conditions 1 and 2 above are as follows.

First, as a major premise, if an index I can actually distinguish QRNGs and PRNGs, its value $I(X)$ must have different probability distributions depending on whether X is from a QRNG or a PRNG. Or equivalently, if $I(X)$ has exactly the same distribution for the two sources, then I will never be able to distinguish them.

Under this premise, condition 1 above demands that there exists a rule D_I that translates such difference in distributions of $I(X)$ to one bit, and that the one bit can be used to classify the sources into two groups. Ideally, D_I should be strong enough to discriminate QRNGs and PRNGs without error, but that is not necessarily required here. It here suffices that the distribution of $D_I(I(X))$ exhibits a slight difference depending on whether X is from a QRNG or a PRNG (see Sec. II B 2 for details).

Condition 2 demands that the computation necessary for such classification can be carried out within a realistic time. For example, if the calculation of a given type of I or D_I takes an infinite, or a finite but unrealistically long time (e.g., longer than the age of the universe even using all the computers on earth), then such I or D_I are considered incalculable in practice, and thus excluded them from consideration.

As to the rigorous notion of "can be calculated in a realistic time," we follow the tradition of computer science and define it as "computable by an algorithm (Turing machine) in time which is a polynomial of the input length" [35]. Throughout the paper, we will often call a polynomial-time algorithm an *efficient* algorithm. Also, unless otherwise stated, whenever we say simply an "algorithm," it refers to an efficient algorithm.

A. Facts on random number generators (RNGs)

1. True random number generator (TRNG)

Throughout the paper, $P(E)$ denotes the probability of an event E.

For each $n \in \mathbb{N}$, we denote by U^n the random variable of the uniformly random n bits; i.e., $U^n \in \{0, 1\}^n$, and $P(U_n = x) = 2^{-n}$ for all $x \in \{0, 1\}^n$. This random variable U^n achieves the maximum Shannon entropy possible for each n,

$$H(U^n) = n \tag{1}$$

[where $H(X)$ denotes the Shannon entropy of a random variable X; $H(X) = -\sum_x p(x)\log_2 p(x)$, $p(x) = P(X = x]$. See, e.g., Ref. [26]). For this reason, we will call U_n the true random number generators (TRNGs).

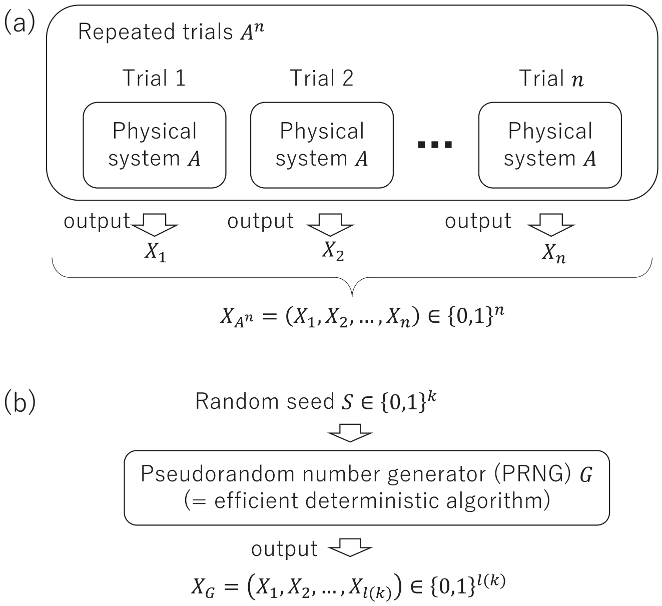
(a)
Repeated trials A^n
Trial 1
Trial 2
Trial n
Physical system A
Physical system A
Physical system A
output
output
output
X_1
X_2
X_n
$X_{A^n} = (X_1, X_2, \dots, X_n) \in \{0,1\}^n$
(b)
Random seed $S \in \{0,1\}^k$
Pseudorandom number generator (PRNG) G (= efficient deterministic algorithm)
output
$X_G = (X_1, X_2, \dots, X_{l(k)}) \in \{0,1\}^{l(k)}$

FIG. 1. (a) Repeated trials A^n of experiments using the same physical system A. (b) Pseudorandom number generator (PRNG) G.

2. TRNGs as a special case of QRNGs

The TRNGs are a special case of QRNGs. That is, the TRNGs can be realized in a quantum system in principle.

The case of $n = 1$, U^1, can be realized because there exists a quantum system A that outputs bit values 0,1 exactly with probability 1/2, which realizes. For example, let A be a quantum system where one (i) generates one of the Z basis states $\{|0\rangle, |1\rangle\}$ and (ii) performs a projective measurement on it using the X basis $\{|\tilde{0}\rangle, |\tilde{1}\rangle\}$, with $|\tilde{0}\rangle := (|0\rangle + (-1)^b|1\rangle)/\sqrt{2}$. Also, if we let A^n denote an n repetition of a system A, then A^n realize U^n for an arbitrary $n \in \mathbb{N}$ [Fig. 1(a)].

At this point, one might argue that in a real physical system there is always a limitation of experimental accuracy, and it might not be possible to achieve probability 1/2 strictly. Even if it were possible, one may not be able to repeat an arbitrary number of trials using exactly the same system, due to limitations in costs and materials.

Still, it should be noted that, in theory, one cannot deny the existence of such A as well as its iterations A^n. That is, no rationale can exclude the TRNG from QRNGs. Hence, for example, if one claims that an index I can distinguish QRNGs from PRNGs, one should also assume that it can distinguish the TRNG, a type of QRNG, from PRNGs.

3. Pseudorandom number generator (PRNG)

Informally, a pseudorandom number generator (PRNG) is

P1: A method for expanding a short random bit string s (called a *random seed*) into a long bit string x

[cf. Fig. 1(b) and Definition 1], where

P2: The generated string x is in some sense similar to a true random bits string (i.e., an output of the TRNG).

Condition P1 means that, once one feeds a short random seed s to a PRNG, one can obtain a virtually inexhaustible number of "random" bits x. PRNGs are useful in this sense in practice, but the drawback is that the resulting x is much less random than TRNG.

Condition P2 means that for certain usages the small randomness of x may not be apparent, and x can be a substitute for the TRNG. For example, in numerical calculations, it is observed that using PRNG instead of TRNG does not significantly change the results.

Condition P1 is stated formally as follows.

Definition 1 (PRNG; cf. Fig. 1(b)). Pseudorandom number generator (PRNG) is a (efficient) deterministic algorithm such that for any $k \in \mathbb{N}$ and any input $s \in \{0,1\}^k$, the result $G(s)$ is a string of length $l(k)$, where $l(k)$ is a polynomial satisfying $l(k) > k$ for $\forall k \in \mathbb{N}$. The polynomial $l(k)$ is called an expansion factor.

Here a deterministic algorithm G is the one in which, once the input value s is determined, the output value x is also uniquely determined.

Since the random seed S is a k-bit long random variable, its entropy is at most k bits. Also, since the algorithm G deterministic, the entropy of its output $X = G(S)$ can at most be that of S. Thus, we have

$$H(G(S)) \leqslant k < l(k), \tag{2}$$

meaning that, unlike in the case of the TRNG, the entropy of PRNG's output $X = G(S)$ can never attain the theoretical maximum $l(k)$ [cf. Eq. (1)]. PRNGs are definitely inferior to the TRNGs in this sense.

4. Cryptographic PRNG (CPRNG)

Next we proceed to the formal definition of condition P2. Unlike condition P1, the definition of P2 differs depending on the academic field. Here we particularly adopt the definition in the field of cryptography, or *cryptology*.

In cryptology, the phrase "in some sense similar" in condition P2 is taken pessimistically and interpreted as "distinguishable by any efficient discrimination algorithm." This is because rationale cryptanalysts will always employ the best attacking method possible. In other words, no cryptanalysts will deliberately choose an inferior method when better methods are available.

From this point of view, condition P2 can formally be stated as follows.

Definition 2 (CPRNG. Ref. [32], *Def. 3.14).* We say that a PRNG G is a cryptographic PRNG (CPRNG) if it satisfies, in addition to the properties of Definition 1, the following: For any (efficient) probabilistic algorithm D, there is a negligible function negl such that

$$\left|P(D(G(U^k)) = 1) - P(D(U^{l(k)}) = 1)\right| \leqslant \mathrm{negl}(k). \tag{3}$$

Several remarks are in order for this definition. First, probabilistic algorithms are those which perform coin tosses internally (see, e.g., [33,35]). Second, a function $f : \mathbb{N} \to \mathbb{R}$ is called negligible if it satisfies: For any positive polynomial (polynomial whose output is positive for any input $\in \mathbb{N}$) $p(k)$, there exists $N \in \mathbb{N}$ such that

$$\forall k \geqslant N,\ f(k) < \frac{1}{p(k)}; \tag{4}$$

see, e.g., Ref. [33], Def. 1.3.5. Finally, the distinguisher algorithm D may vary depending on the PRNG algorithm G; in other words, this definition covers the situation where the cryptanalysts who try to distinguish between the PRNG and the TRNG *know* the PRNG algorithm.

In fact, however, no explicit construction of CPRNG is known, which satisfies the properties of Definition 2. No proof of its existence has been given either. Hence the existence of a CPRNG is a mere assumption, not a proven fact.

In such a situation, what is mostly done in cryptology is either (i) to mathematically prove the existence of a CPRNG from other assumptions or (ii) to mathematically prove the security of a cryptosystem having G as a component, assuming that G is a CPRNG; see, e.g., Refs.[32,33]. As a result, most of the results obtained so far in cryptology do not hold without the existence of a CPRNG or other stronger assumptions, such as the existence of a one-way function.

It may be easier to understand this situation by considering CPRNG in cryptology as an analog of the second law in thermodynamics. In thermodynamics, there are two situations similar to (i) and (ii) above, namely, (i′) if one accepts the second law as a hypothesis, one can prove useful results including the existence of entropy, and (ii′) if one accepts the principles of statistical dynamics as hypotheses, one can explain to some extent (though one cannot prove) why the second law holds.

Hence, although the existence of a CPRNG is nothing more than a hypothesis, if any result contradicting with it were obtained, it would immediately mean the collapse of cryptology as a whole, and the impact would be enormous.

In terms of practicality, this can be stated as follows. Many standardized cryptographic schemes that we currently use specify a PRNG to be employed as their component. For example, a cryptographic scheme called ChaCha20-Poly1305 employs a PRNG called ChaCha20 [32,36]. When theoretically guaranteeing the security of these standardized cryptographic schemes, it is customary to simply assume (rather than prove) that the PRNG is a CPRNG (or a pseudorandom function, which can be utilized to construct a CPRNG; see Sec. 3.5.1, Ref. [32]), and then to "prove" that the scheme using it is secure based on the assumption (see, e.g., Sec. 3.2, Ref. [32]). Therefore, if the existence of a CPRNG were denied, the basis for theoretically guaranteeing the security of standardized cryptographic schemes would be lost.

B. Minimal requirements on randomness measures

References [27,30] discuss randomness measures for distinguishing QRNGs and PRNGs. There a randomness measure I is an index $I(X)$ that can be calculated from the output bit string $X \in \{0,1\}^n$ of a RNG, and reflects the randomness of the RNG. However, they did not necessarily discuss conditions for I to be useful in practice. Below will discuss such conditions.

1. Efficiently computable randomness measure

Some indices I, such as LZ complexity (see Sec. III B 1 a), take a discrete value, while others, such as Borel normality (see Sec. III B 1 b), take a continuous value. Those indices I of discrete value can of course be represented by a finite number of bits. On the other hand, those of a continuous value cannot be calculated as it is in finite time, hence one needs to adapt

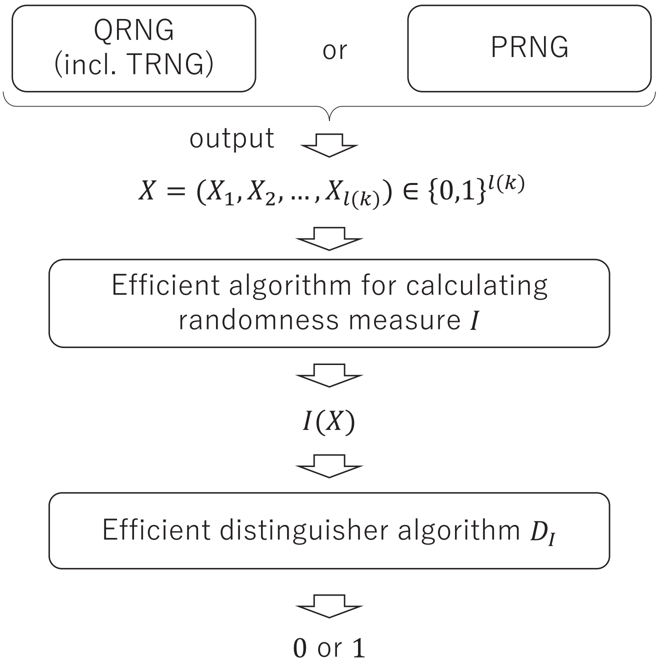
QRNG
(incl. TRNG)
or
PRNG
output
$X = (X_1, X_2, \dots, X_{l(k)}) \in \{0,1\}^{l(k)}$
Efficient algorithm for calculating randomness measure I
$I(X)$
Efficient distinguisher algorithm D_I
0 or 1

FIG. 2. Setting for discriminating QRNGs (including the TRNG) from PRNGs using a randomness measure I and a distinguisher D_I corresponding to I.

them to a finite number of bits, e.g., by truncating the value to a certain number of significant digits. As a result, any index I needs to be of finite bit length.

Moreover, in order for the index I to be calculable not only in finite time but also in a realistic time, we need an efficient algorithm that can compute it. Hence we need the following condition on I.

Definition 3 (Efficiently computable measure I; cf. Fig. 2). A randomness measure is an efficient probabilistic algorithm $I : \{0, 1\}^* \to \{0, 1\}^*$.

Here $\{0, 1\}^*$ denotes the set of all bit strings of a finite length, $\{0, 1\}^* = \bigcup_{n \in \mathbb{Z}_{\geq 0}} \{0, 1\}^n$ (see, e.g., Refs. [32,33]).

Note that Definition 3 can accommodate the case where one handles multiple indices $I_1, I_2, \dots$ simultaneously: Let I be a concatenating $I_1, I_2, \dots$, for example.

2. *Efficient distinguisher for a randomness measure*

Furthermore, for I to be an effective randomness measure, it is necessary that "I reflects the randomness of the RNG." In this paper we interpret this property as that "the value $I(X)$ has different probability distributions depending on whether the source of X is a TRNG or a PRNG." Indeed, if this is not the case, then $I(X)$ will have the same distribution irrespective of the randomness of X's source. It is apparent that such $I(X)$ conveys no information on the randomness of X.

In addition, in order for humans to be able to exploit such differences in probability distribution, a quantitative criterion is necessary. That is, one needs an algorithm D_I which determines from the value of $I(X)$ whether the randomness of X's source is good or bad (0 or 1). Also, in order for D_I to be calculable in a realistic time, D_I needs to be efficient.

In summary, for any effective randomness measure I, there should exist an algorithm D_I of the following type.

Definition 4 (Distinguisher D_I for measure I; cf. Fig. 2). A disting-uisher for a randomness measure I is an efficient probabilistic algorithm $D_I : \{0, 1\}^* \to \{0, 1\}$.

Also, we say that I is a good randomness measure if the distribution of $D_I(I(X))$ is significantly different depending on whether the source of X is a QRNG or a PRNG.

Under these definitions, the ideal situation is, of course, where the algorithm $D_I \circ I$ can discriminate QRNGs and PRNGs perfectly without any error. However, it should be noted that we do not necessarily require such ideal discrimination here.

For example, I can be regarded a good randomness measure as long as the result $D_I(I(x)) = 1$ is reliable, even if $D_I(I(x)) = 0$ is not. This corresponds to the following common situation in randomness tests (such as Ref. [37]): If x fails the test [i.e., if $D_I(I(x)) = 1$] one can safely conclude the source of x is a bad one (in our case, a PRNG), but even if x passes the test [i.e., if $D_I(I(x)) = 0$] it does not necessarily mean that the source was a good one (in our case, a QRNG).

C. Main result: A good randomness measure is not feasible

In fact, the existence of a good randomness measure I directly contradicts the common assumption of cryptology. Hence a good randomness measure is not feasible.

More precisely, from what we have already discussed, one can conclude that

(i) If a cryptographic PRNG (CPRNG), defined in Sec. II A 4, exists, it is indistinguishable from the TRNG.

(ii) If there exists the algorithm $D_I \circ I$ that can distinguish QRNGs and PRNGs, it should also be able to distinguish the TRNG (a type of QRNG) and CPRNGs (a type of PRNGs).

However, these statements are clearly contradictory, thus one must give up either one. That is, one needs to accept either of the followings statements,

(i′) A CPRNG does not exist.

(ii′) No algorithm $D_I \circ I$ corresponding to any randomness measure I can distinguish between the TRNG (a type of QRNG) from CPRNGs (a type of PRNGs).

Recall that, as mentioned in the second half of Sec. II A 4, the existence of CPRNG is one of the most common assumptions in cryptology, on which most results of the field are based. Thus, choosing (i′) is equivalent to making a very challenging claim that denies much of cryptology.

Therefore, from a conservative standpoint, statement (ii′) should be accepted. The rigorous form of this statement is as follows.

Theorem 1. Suppose that there exists a CPRNG G, then for any randomness measure I and any distinguisher D_I corresponding to it, we have

$$\left|P(D_I(I(G(U^k))) = 1) - P(D_I(I(U^{l(k)})) = 1)\right| \leqslant \mathrm{negl}(k), \tag{5}$$

where $l(k)$ is the expanding factor of G. [Recall that $P(E)$ denotes the probability of an event E.]

Proof. This theorem follows trivially from Definition 2 of CPRNG. Indeed, for any given I and D_I satisfying Definitions

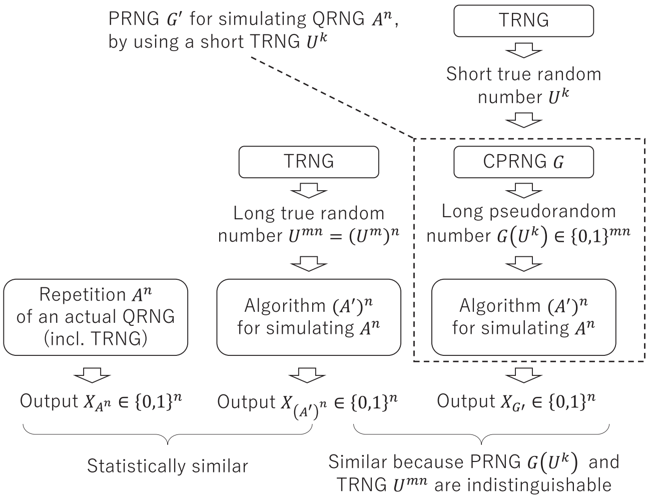
PRNG G' for simulating QRNG A^n, by using a short TRNG U^k
TRNG
Short true random number U^k
TRNG
Long true random number $U^{mn} = (U^m)^n$
CPRNG G
Long pseudorandom number $G(U^k) \in \{0,1\}^{mn}$
Repetition A^n of an actual QRNG (incl. TRNG)
Algorithm $(A')^n$ for simulating A^n
Algorithm $(A')^n$ for simulating A^n
Output $X_{A^n} \in \{0,1\}^n$
Output $X_{(A')^n} \in \{0,1\}^n$
Output $X_{G'} \in \{0,1\}^n$
Statistically similar
Similar because PRNG $G(U^k)$ and TRNG U^{mn} are indistinguishable

FIG. 3. Construction of PRNG G' that simulates the output of QRNG A, which is a biased bit.

3 and 4, the algorithm $D_I \circ I$ becomes an example of the distinguisher algorithms D for G, defined in Definition 2. Thus, we have (5). ■

We note that a similar theorem as Theorem 1 holds by assuming the existence of a one-way function, not of a CPRNG. This is because a CPRNG can be constructed from one-way functions (see, e.g., Ref. [33], Sec. 3.4).

D. Generalization of Theorem 1

In the previous section, we assumed that QRNG A outputs a bit 0,1 with exactly probability 1/2; that is, we consider only the case where A realizes the one-bit TRNG U^1. In addition, the expansion factor $l(k)$ of the CPRNG G that mimics $A^{l(k)}$ could not be chosen freely. Below we relax these restrictions and generalize Theorem 1.

1. A simple case (biased random bit generators)

First, we outline the idea with a simple example.

Suppose that QRNG A output 1-bit $X \in \{0, 1\}$ as in the previous section, but the probability may be biased; i.e., $p = P(X = 1)$ may not be 1/2. In this case too, essentially the same result as Theorem 1 holds: If there exists a CPRNG G, then one can construct a PRNG G' that simulates $A^{l(k)}$, with an arbitrary polynomial $l(k)$ being the expansion factor. In addition, the outputs of $A^{l(k)}$ and G' are indistinguishable by using any I and D_I.

The rigorous proof of this claim will be given in the next section and in Appendix B after generalizing the setting further. Below we temporarily give a proof sketch for the simple case above (cf. Fig. 3).

First, note that A can be efficiently simulated with accuracy 2^{-m} in probability, by using a deterministic algorithm A' whose input is the m-bit long TRNG U^m. For example, (i) approximate probability p by a fraction $p = b/2^m, b \in \mathbb{N}$, and (ii) let A output 1 if the value of U^m is less than or equal to b, and 0 otherwise.

Next let $(A')^n$ denote the n repetition of A', Clearly, $(A')^n$ is a deterministic algorithm that simulates A^n by using the input of nm-bit TRNG U^{mn}.

If we then replace the input U^{mn} of $(A')^n$ with the output of CPRNG $G(U^k)$ (having an expansion factor $l(k) = mn$), then we obtain the desired PRNG G' that mimics A^n.

In this setting, the output of G' is indeed indistinguishable from the output of $A^{l(k)}$, because: (i) By choosing m sufficiently large, the output $X_{(A')^n}$ of $(A')^n$ can approximate the output X_{A^n} of A^n with arbitrarily small error probability. That is, X_{A^n} and $X_{(A')^n}$ are indistinguishable. (ii) The TRNG U^{mn} and the CPRNG $G(U^k)$ are indistinguishable by the definition of CPRNG. Thus, the outcome of an algorithm $(A')^n$ on their input, namely $X_{(A')^n}$ and $X_{G'}$, are also indistinguishable.

2. Efficiently classically simulatable systems

Next, we generalize the settings of the previous sections further and discuss them rigorously.

The generalized settings are as follows. The output X of a QRNG A is no longer limited to one bit, but can be any finite set $\mathcal{X}$ (i.e., $X \in \mathcal{X}$). Then we require that A be efficiently classically simulatable [34]. In recent and fashionable terminology, this means that the system A does not exhibit quantum supremacy. The precise definition is as follows.

Definition 5 (Efficient classical simulation). (Ref. [34], *Def.* 2) Suppose that a physical system A outputs a random variable $X \in \mathcal{X}$. An efficient classical simulation for A is an efficient probabilistic algorithm A' which, on input 1^n (sequence of n symbols of "1"), outputs $\in \mathcal{X}$ that satisfies

$$\|P(X_A) - P(X_{A'(1^n)})\|_1 \leqslant 2^{-n}. \tag{6}$$

We say that A is efficiently classically simulatable (ECS) if such A' exists.

Several technical remarks are in order related to this definition: First, the symbol $\|P(X) - P(Y)\|_1$ denotes the total variational distance between the probability distributions of random variables $X, Y \in \mathcal{X}$; $\|P(X) - P(Y)\|_1 := \frac{1}{2}\sum_{x\in\mathcal{X}} |P(X = x) - P(Y = x)|$. Second, the input 1^n is meant to specify parameter n and to require that A' halts within a polynomial time of n. Third, in the original definition (i.e., in Definition 2 of Ref. [34]) A' has an additional input besides 1^n, but we here omit it because in this paper we restrict ourselves to the case where physical system A is a QRNG and thus has no input. Finally, in the statement of Definition 5, we used the term "physical system" instead of "quantum system" because the discussions below hold not only for quantum systems, but for physical systems in general.

Additionally, to comply with the fact that the output of A is a $\mathcal{X}$ string rather than a bit string, we let the input to index I also be an $\mathcal{X}$ string. That is, we let I be an efficient probabilistic algorithm $I : \mathcal{X}^* \to \{0, 1\}^*$.

Under these settings, we have the following generalization of Theorem 1.

Theorem 2. Suppose that there exists a CPRNG. Then for any ECS physical system A, and for any positive and increasing polynomial $l(k)$, there exists a deterministic algorithm $B : \{0, 1\}^* \to \mathcal{X}^*$ which, on input a k-bit string, outputs a $\mathcal{X}$ string of length $l(k)$ that satisfy

$$|P(D_I(I(B(U^k))) = 1) - P(D_I(I(X_{A^{l(k)}})) = 1)| \leqslant \mathrm{negl}(k). \tag{7}$$

for any pair of a randomness measure I and a distinguisher D_I.

In other words, for any ECS system A, there exists an algorithm B that impersonates it with low randomness. Moreover, the output size $l(k)$ of $B(U^k)$ can be chosen to be an arbitrary polynomial.

The proof of Theorem 2 is given in Appendix B.

III. EMPIRICAL ANALYSES

In this section, we give empirical examples to support Theorems 1 and 2. For this, we perform the data analyses on two kinds of QRNGs: quantum coin toss games and the Bell tests. We start this section by presenting the details of those data sets.

A. Data

1. *Quantum coin toss games*

In quantum coin toss games, a type of QRNGs, we first prepare the initial state $|\Psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, where $\{|0\rangle, |1\rangle\}$ is an orthonormal basis of the state space $\mathbb{C}^2$ of a two-level system. We then perform projection measurement on the initial state $|\Psi\rangle$ along the computational bases $\{|0\rangle, |1\rangle\}$ and find the outcome $a \in \{0, 1\}$. Repeating this process N times, we obtain a bit string $a^N = (a_1, a_2, \ldots, a_N)$, where $a_i \in \{0, 1\}$.

In IBM Quantum, the initial state is restricted to the computational basis state $|0\rangle$. The state $|\Psi\rangle$ is implemented as the application of the Hadamard gate $H = (|0\rangle\langle 0| + |0\rangle\langle 1| + |1\rangle\langle 0| - |1\rangle\langle 1|)/\sqrt{2}$ to the computational basis state $|0\rangle$. Therefore, we can rewrite the quantum coin toss as the following three steps: (i) state preparation in the computational basis $|0\rangle$ for a two-level system (qubit) in a quantum processing unit (QPU), (ii) application of the Hadamard gate H, and (iii) projection measurement along the computational bases $\{|0\rangle, |1\rangle\}$.

Following the above steps, we made a data set by using `ibmq_manila` provided by IBM Quantum. The data set was made on February 23, 2023, and consists of 100 bit strings with the length N aligned to $N = 20\,000$.

To make a comparison between this QRNG and PRNGs, we performed classical simulations using three types of PRNGs. The first and second types are `ibmq_qasm_simulator` and `imbq_qasm_simulator` with the device noise model of `ibmq_manila`, respectively. The third is ChaCha20 algorithm described in Refs. [32,36]. Note that the simulators are regarded as PRNGs since they run on classical computers and use a short random seed. We hereafter specify data sets by the name of the device used for their data acquisition process.

Table I shows the mean relative frequency of bit value 1 in the entire bit string. The numbers after ± are the standard deviation obtained after 100 trials. Note that the mean relative frequency of `ibmq_qasm_simulator` is closer to the theoretical value 1/2 than that of `ibmq_manila`, and these two values are mutually distinguishable within the standard deviations. Note also that, in contrast, the mean relative frequency of `ibmq_qasm_simulator` (with noise) is comparable to that of `ibmq_manila` within the standard deviations.

TABLE I. Mean relative frequency of bit value 1 and standard deviation of three data sets of 100 bit strings with length $N = 20\,000$ created by the quantum coin tosses. The numbers after ± are the standard deviation obtained after 100 trials.

Device used in data acquisition	Mean relative frequency of bit value 1
`ibmq_manila`	0.4851 ± 0.0034
`ibmq_qasm_simulator`	0.4999 ± 0.0034
`ibmq_qasm_simulator` (with noise)	0.4893 ± 0.0036

2. *Innsbruck experiment*

To examine observation (ii) in the Introduction, as an example of the actual Bell tests, we employ the data set taken for [4], which reports the first experiment of the Bell test free from the locality loophole. We hereafter call the series of the Bell tests conducted in [4] the Innsbruck experiments.

In the Innsbruck experiments, entangled photon pair $|\Phi\rangle = (|0\rangle|1\rangle - |1\rangle|0\rangle)/\sqrt{2}$ is created. Each photon in the pair is distributed to space-like separated two points (hereafter called Alice and Bob, respectively), but some photons are lost during the distribution process. For the distributed photons in each distribution process, Alice performs a measurement by choosing the measurement axis x out of two alternatives $x \in \{0, 1\}$, and obtains the outcome $a \in \{0, 1\}$. Bob also does so for each photon. We hereafter denote Bob's measurement axis and outcome by $y \in \{0, 1\}$ and $b \in \{0, 1\}$, respectively.

Repeating the above measurement processes many times, they store the data of the measurement outcomes with their time stamps and the records of the measurement axes. We call each of these records of the series of measurements a sample. In particular, we shall analyze 21 samples, each named longdist*, which are the samples actually used in the Innsbruck experiments [31].

Given a sample, we extract a pair of *coincident outcomes*, namely, those detection events that occur simultaneously on both Alice's and Bob's sides within the threshold given in [31] with the time stamps difference. We denote the outcomes by (a, b), and the number of the pairs of the coincident outcomes in the sample by N. We construct three bit strings, that is, Alice's bit string $a^N = (a_1, a_2, \ldots, a_N)$, Bob's bit string $b^N = (b_1, b_2, \ldots, b_N)$, and a mixed bit string $c^N = (a_1, b_1, a_2, b_2, \ldots, a_N, b_N)$.

By construction, we may expect that the bit string c^N is less random than a^N and b^N, if the given sample violates the Bell inequality. For every pair (a_i, b_i) in c^N has a strong correlation not described by any local hidden variable theories (LHVTs), and thereby c^N becomes more structured than the bit strings expected from LHVTs. This inherent difference between c^N and the others is also examined in this paper.

3. *Efficient simulation using ChaCha20*

To give numerical support to Theorems 1 and 2, by using the PRNG called ChaCha20 algorithm [32,36], we made the pseudorandom number strings that mimic the bit strings obtained from the above two experiments by following the procedure given in Sec. II D 1 for the biased random bit sequences.

More precisely, for the Innsbruck experiment, we used the following process. First given the relative frequency and length of an experimental bit string, we generated a pseudorandom number byte string having twice the length of the given bit string. Then we convert each two bytes of the pseudorandom number byte string to a real number. We compare each of these real numbers with a threshold value, which is the relative frequency value rescaled by 2^{16}, and convert it to a bit 0 (1) if it is higher (lower) than the threshold.

We also performed the same procedure for the quantum coin toss games by using the mean relative frequency of the data taken by `ibmq_manila`.

B. Measures

We characterize the bit strings in the data sets described in the previous section by the (algorithmic) randomness. In addition to this, for the data set of the Innsbruck experiment, we measure how much the data are nonlocal. We hereafter introduce the measures of algorithmic randomness and nonlocality.

1. Randomness measures

a. Lempel-Ziv (LZ) complexity. Let us suppose that we are given a bit string $s^N = (s_1, s_2, \dots s_N) \in \{0,1\}^N$, and consider the following parsing:

$$E(s^N) = s(1, h_1)s(h_1+1, h_2)\cdots s(h_{m-1}+1, N), \tag{8}$$

where $s(i, j) = (s_i, s_{i+1}, \dots, s_j)$ is the substring of s^N from the ith bit to the jth. Now let us set $h_1 = 1$ and make the parsing in such a way that $s(h_{i-1}+1, h_i)$ is the minimal substring that has not already appeared as the substring $s(h_{k-1}+1, h_k)$ with $k < j$. Then it has been known that the $E(s^N)$ is uniquely determined if we follow the above rule of parsing [27].

Let $c(N)$ be the number of the substrings constructed in the above rule. Then $c(N)$ has an upper bound [27]

$$c(N) < \frac{N}{(1-\varepsilon_N)\log_2(N)}, \tag{9}$$

where

$$\varepsilon_N = 2\frac{1+\log_2\log_2(2N)}{\log_2(N)}. \tag{10}$$

Note that inequality (9) follows solely from the definition of $c(N)$, and thereby holds for any bit string s^N of length N.

Motivated by this upper bound of $c(N)$, we define

$$K(N) = c(N)\frac{\log_2 N}{N}, \tag{11}$$

which has been widely used as a measure of algorithmic complexity in literature [21–25,28,29]. We hereafter call $K(N)$ the LZ complexity in this paper, although much literature calls $K(N)$ the normalized LZ complexity.

As suggested in inequality (9), the upper bound of the LZ complexity $K(N)$ depends on the length N. To make it reasonable that the comparison of the algorithmic complexity between the bit strings with mutually distinct lengths, let us normalize the LZ complexity $K(N)$ by the maximal value of $c(N)$ for the given N. For this, we recall that the maximal value of $K(N)$ is attained by the bit string whose substrings are exhaustively aligned and lengthened from left to right, for example,

$$t^N = (0, 1, 0, 0, 0, 1, 1, 0, 1, 1, \dots), \tag{12}$$

whose parsing is given as

$$\begin{aligned} t(1,1) &= (0), & t(2,2) &= (1),\\ t(3,4) &= (0,0), & t(5,6) &= (0,1), \end{aligned} \tag{13}$$

and so on. In this case, the length of the bit string composed of all the mutually distinct bit strings whose lengths are up to m is given by

$$l_m = \sum_{k=1}^{m} k2^k = (m-1)2^{m+1} + 2. \tag{14}$$

Let m^* be the number to minimize the difference $N - l_m \geqslant 0$. Then we obtain the number of the bit strings to compose t^N as

$$c_{\max}(N) = \sum_{k=1}^{m^*} 2^k + \left\lceil \frac{N - l_{m^*}}{m^*+1} \right\rceil = 2^{m^*+1} - 2 + \left\lceil \frac{N - l_{m^*}}{m^*+1} \right\rceil, \tag{15}$$

where $\lceil x \rceil$ is the minimal integer exceeding x. It follows that the maximum of $K(N)$ is given as

$$K_{\max}(N) = c_{\max}(N)\frac{\log_2 N}{N}. \tag{16}$$

Based on the above argument, we define the normalized LZ complexity

$$\kappa(N) = \frac{K(N)}{K_{\max}(N)} = \frac{c(N)}{c_{\max}(N)}. \tag{17}$$

To close this introduction of the LZ complexity, let us point out the inconsistencies appearing in data analyses of Ref. [24]. Using the fact that ε_N decreases monotonically with N and that $\varepsilon_{4096} < 0.784$ holds, we obtain

$$c(N) < 4.6\frac{N}{\log_2(N)} \quad \text{for} \quad 4,096 \leqslant N. \tag{18}$$

Thus for the normalized complexity measure $K(N)$,

$$K(N) < 4.6 \quad \text{for} \quad 4,096 \leqslant N \tag{19}$$

must always hold.

In contrast, almost half of all trials shown in Table I of Ref. [24] violate the upper bound (19). More precisely, for 16 of 37 trials in Table I (namely, longdist X with $X = 1, \dots, 4$, 10, 11, 13, 20, 23, 30, 32, 33, 34, 36 and 37, and Conlt3), the authors conclude the $K(N) > 7$ and $N > 4,096$ hold simultaneously, but this clearly conflicts with inequality (19), implying inaccuracy in their data processing.

b. Borel normality measure. The Borel normality measures the distribution of substrings in the given bit string [30]. Let $B_m = \{0,1\}^m$ be the set of the bit strings with length m, and $N_j^m(s^N)$ be the number of occurring the lexicographical jth bit string of length m in s^N. Let us denote the length of s^N over B_m by $|s^N|_m$ and define $|s^N| = |s^N|_1$. Then the bit string

s^N is Borel normal (with accuracy $\frac{1}{\log_2 |s^N|}$) if we have

$$\left| \frac{N_j^m(s^N)}{|s^N|_m} - \frac{1}{2^m} \right| \leqslant \frac{1}{\log_2 |s^N|}, \tag{20}$$

for every integer $1 \leqslant m \leqslant \lfloor \log_2 \log_2 |s^N| \rfloor$ and $1 \leqslant j \leqslant 2^m$ [30].

From the above definition of the Borel normality, the following measure was proposed in [25]:

$$B(s^N) = \max \left| \frac{N_j^m(s^N)}{|s^N|_m} - \frac{1}{2^m} \right| \log_2 |s^N|, \tag{21}$$

where the maximum is over $m = 1, 2, \dots, \lfloor \log_2 \log_2 |s^N| \rfloor$ and $j = 1, 2, \dots, 2^m$. The measure $B(s^N)$ ensures that the bit string s^N is Borel normal if $B(s^N) \leqslant 1$.

2. *Non-locality measure*

We first introduce the CHSH correlation function [38]

$$S = \left| \sum_{x,y} (-1)^{xy} [P(A = B|xy) - P(A \neq B|xy)] \right|, \tag{22}$$

where $P(A = B|xy)$ is the conditional probability of the outcome coincidence given the pair of the measurement axes (x, y) and $P(A \neq B|xy)$ is that of the noncoincident outcomes. The CHSH correlation function has the upper bound $S \leqslant 2$ for any LHVTs [38] whereas for quantum mechanics it exceeds the bound and attains $2\sqrt{2} \approx 2.828$ [39]. We employ the CHSH correlation function as the measure of nonlocality.

C. Results of the empirical analyses

1. *Quantum coin tosses*

We discuss whether there is a difference in the randomness of the bit strings created by a QRNG (quantum computer) and PRNGs including simulators by considering the simplest setup, i.e., quantum coin toss games. Figure 4 shows the randomness measures (the algorithmic complexity measures mentioned above) of our data sets.

As to the (normalized) LZ complexity, the bit strings obtained from the simulator with the device noise model and ChaCha20 are comparable to those obtained from the QRNG. The bit strings from the simulator (without the device noise model) have a slightly higher algorithmic complexity than those from the QRNG. This is presumably because the mean relative frequency of the data obtained from the simulator is almost 0.5, which is significantly higher than that of the data set from the actual device.

In contrast, the Borel normality measure distinguishes all the cases.

Let us look at the above observation by using the hypothesis testing (Welch's t test) on whether the simulations have no difference from the actual device: For the case of noiseless simulation against the actual device, we found the statistic $t \approx -3.01$ with its p value being 2.96×10^{-3} for the LZ complexity and normalized one, whereas $t \approx 20.69$ with its p value being 2.15×10^{-51} for the Borel normality measure. This suggests that these two ways of creating the random bit strings are distinguishable from each other if we focus on the average values of the randomness measures.

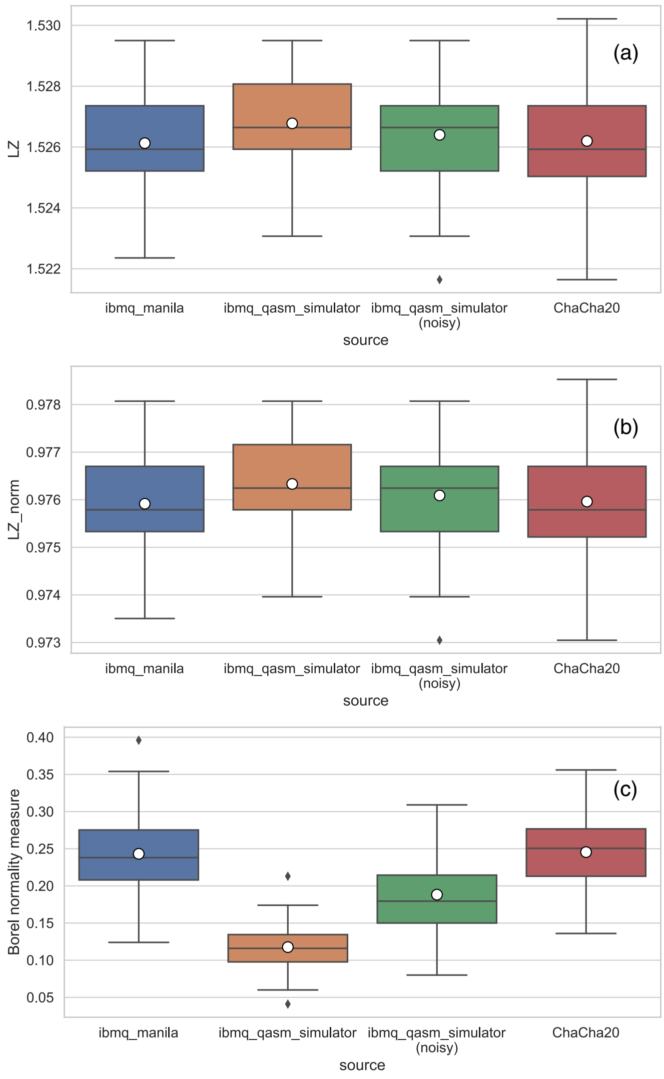

FIG. 4. Box plots of the randomness measures for the quantum coin tosses: (a) the LZ complexity, (b) the normalized LZ complexity, and (c) the Borel normality measure. Throughout this paper, we draw the box-and-whisker plots as follows. Given the data and their attributes to be plotted, the length of the upper (lower) whisker is set as a maximum (minimum) value not exceeding 1.5 times the interquartile range (IQR, the height of the box). Data whose attribute values are outside this range are called outliers, which are plotted as dots.

In contrast, for the case of simulation with the device noise model against the actual device, we observed the statistic $t \approx -1.25$ with its p value being 0.213 for the LZ complexity and normalized one, while $t \approx 7.58$ with its p value being 1.33×10^{-12} for the Borel normality measure. This shows that these two ways of creating the random bit strings are indistinguishable from each other if we employ the (normalized) LZ complexity as shown in Theorem 1, if their relative frequency is comparable. Moreover, for the case of PRNG, the statistic is $t \approx -0.32$ and its p value is 0.749 for the LZ complexity and normalized one, while $t \approx -0.33$ with its p value being 0.739 for the Borel normality measure. Thus, the bit strings created by ChaCha20 are indistinguishable for all the complexity measures we have employed, supporting our theorems.

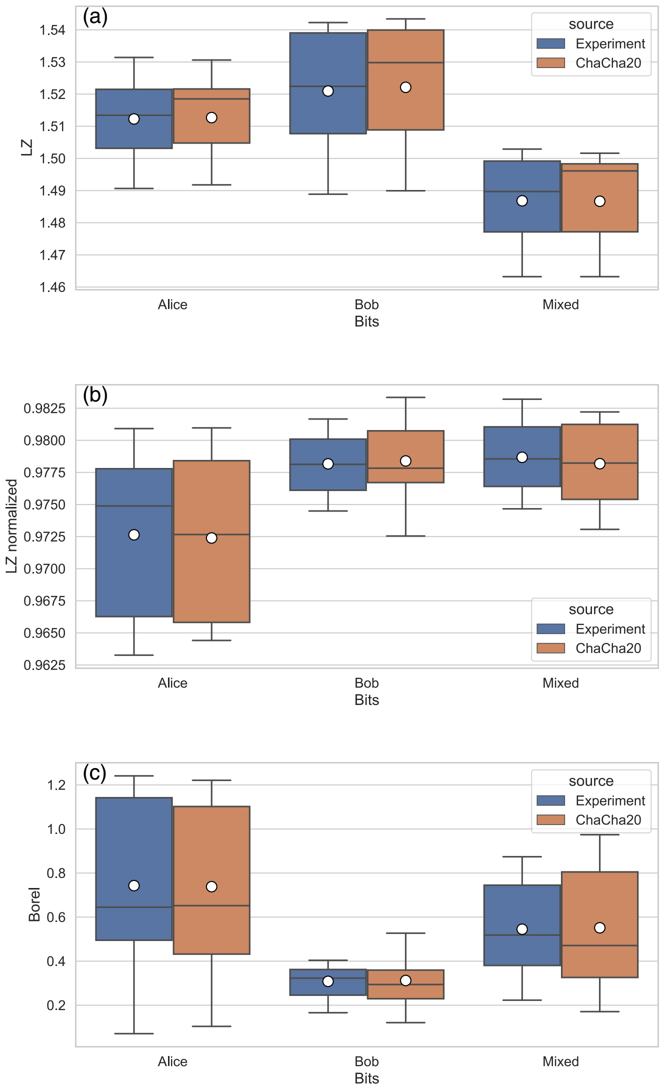

FIG. 5. Box plots of the randomness measures for the bit strings generated from the Innsbruck experiment and ChaCha20: (a) the LZ complexity, (b) the normalized LZ complexity, and (c) the Borel normality measure. The outlier coming from longdist4 is omitted. The bit strings at the Bob's station are more random than those at the Alice's station. The mixed bit strings are less random if evaluated by the LZ measure.

2. Innsbruck experiment

Motivated by our theorems and the discussion presented in Sec. III B, we proceed to the analyses of the Innsbruck experiment, or the longdist data set. More concretely, we shall show the descriptive statistics of the randomness measures mentioned above and their correlation coefficients.

In our analyses, we have omitted the longdist4 sample as an outlier due to its relatively small number of the coincident events N; see Appendix A.

Figure 5 shows the descriptive statistics of the randomness measures. All the randomness measures show that Alice's bit strings tend to be more random than Bob's ones. This suggests asymmetry between Alice's apparatus and Bob's due to, e.g., systematic errors since the ideal Bell test has the exchange symmetry between Alice and Bob. Note that one cannot extract information about such asymmetry by using S, which quantifies only the correlation between Alice and Bob. Note that the maximal value of the LZ complexity in our analysis is in accordance with our theoretical consideration in Sec. III B 1 a. More precisely, all our analyses satisfy $K(N) \lesssim 1.55$, which is accordance with $K(N) < \frac{1}{1-\epsilon_N} \approx 2.652$ implied by inequality (9) for the bit length 90 000, an upper bound length of the mixed bit string c^N in our coincidence data.

Above we analyzed Alice and Bob individually, but we also analyzed their mixed bit strings. We found that the relationship between the randomness of Alice's (or Bob's) individual bits and that of the mixed bit sequence appears to be different depending on the measure. This may reflect the fact that the randomness measures we use, although normalized, is still weakly dependent on length. The mixed bit string is statistically the least random if measured with the LZ complexity, whereas the most random with the normalized LZ complexity. Moreover, the Borel normality suggests that its randomness is in between.

We note that the mixed bit sequence should be the least random in the algorithmic sense, since $S > 2$ for many of the longdist data suggests that every bit pair (a_i, b_i) in the mixed bit strings has the strong correlation not described by any LHVTs, resulting in a decrease of the algorithmic randomness. This suggests that perhaps the LZ complexity most captures the latent correlations inherent in the mixed bit strings obtained from the Bell tests.

The data obtained from ChaCha20 successfully imitate those from the Innsbruck experiment. More precisely, the interquartile range (IQR, the box heights in the box-and-whisker plots) are almost the same as those of the associated Innsbruck data.

In the rest of this subsection, we reexamine the anticorrelation suggested in [24]. We first check the correlation between the entropy and the (normalized) LZ complexity. Figure 6 shows that given the conditional probability distributions $p(a|x)$, $p(b|y)$, their entropies

$$H(A|X=i) = -\sum_a p(a|X=i)\log_2 p(a|X=i),$$
$$H(B|Y=i) = -\sum_b p(b|Y=i)\log_2 p(b|Y=i) \qquad (23)$$

for $i = 0, 1$ correlate with the (normalized) LZ complexity. This suggests that the LZ complexity can be understood in terms of entropy. Moreover, we observe that the entropy for the case $x = 0$ and $y = 0, 1$ ranges from 0.98 to 1, whereas that for $x = 1$ does from 0.95 to 1. Since the initial state in Weihs' experiment is the singlet state, its reduced density matrix is completely mixed, leading to the equal weight probability distribution. Therefore, the large deviation of the entropy when $x = 1$ implies the existence of (systematic) errors in Alice's station.

The above correlation can be observed in the entropy of the joint probability distribution

$$H(AB|X=i, Y=j) = -\sum_{a,b} p(ab|X=i, Y=j) \times \log_2 p(ab|X=i, Y=j). \qquad (24)$$

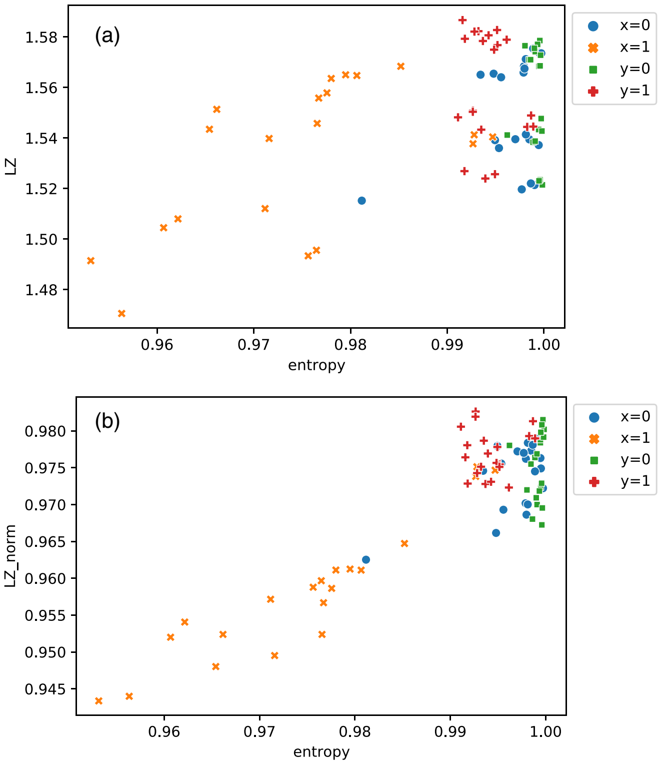

FIG. 6. Scatter plots between the randomness measures and the entropy: (a) the LZ complexity and (b) the normalized LZ complexity. The outlier coming from longdist4 is omitted. The bit strings at the Bob's station are more random than those at the Alice's station.

Indeed, Fig. 7 shows a correlation relation stronger than that shown in Fig. 6. Therefore we can conclude that the LZ complexity is explainable by the entropy of joint probability distribution for the Innsbruck experiment.

In contrast to entropy, we cannot say an anticorrelation exists between nonlocality and randomness. To argue this in detail, recall that the maximum of the LZ complexity depends on N as shown in Sec. III B, and thereby the possible confounder is the number of the coincidence events N. From this observation, we evaluated the Pearson correlation coefficients r between S and the randomness measures with the use of the stratification of N into the following three classes: (i) $N < 20\,000$, (ii) $20\,000 \leqslant N < 40\,000$, and (iii) $N \geqslant 40\,000$.

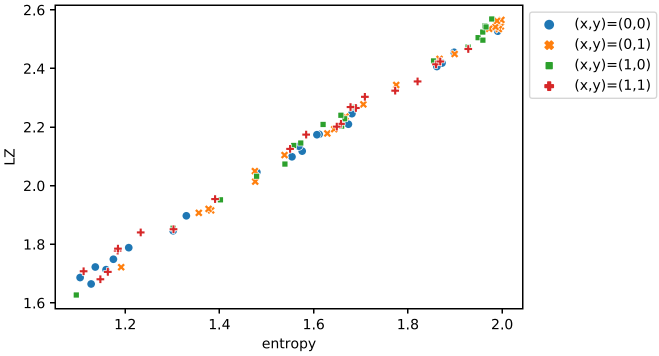

FIG. 7. Scatter plot between the randomness measures and the entropy of the joint probability distribution. The outlier coming from longdist4 is omitted.

Table II shows the correlation coefficients r with its p values and 95% coincidence intervals. We find that $r > 0$ for the (normalized) LZ complexity in general whereas $r < 0$ for the Borel normality except for the mixed bit sequence with $20\,000 \leqslant N < 40\,000$. This is consistent with the fact that the (normalized) LZ complexity becomes larger when the given sequence is more random, whereas the Borel normality is becoming smaller. Besides, we observe that the correlation coefficient for Alice's bit strings is comparable with that for the mixed bit strings, except in the case $N \geqslant 40\,000$. Thus, we can conclude that the degree of the randomness (weakly) correlates with the nonlocality for Alice's bit strings for $N < 20\,000$ and $20\,000 \leqslant N < 40\,000$. This also holds for the mixed bit strings for $N \leqslant 20\,000$. These observations are inconsistent with [24]. On the other hand, all the p values are more than 0.05, implying that it is hard to deduce the anticorrelation for the parent population if we see the longdist data set as samples thereof. Indeed, all the 95% coincidence intervals of r contain $r = 0$.

It is of important to find the typical sample size to determine whether the nonlocality (anti)correlates with the degree of randomness in the parent population. For example, let us take $r = 0.45$, which is the correlation coefficient between S and the LZ complexity of Alice's bit strings for $N \leqslant 20\,000$. According to the standard procedure given in [40], the sample size is 36 to test the hypothesis $r = 0.45$ with significance level $\alpha = 0.05$ (two-tailored) and power $1 - \beta = 0.8$. Moreover, sample size estimation with a pre-specified confidence interval requires a larger sample size: we need the sample size 64, 111, and 247 to test $r = 0.45$ with desired 98% confidence interval half-width 0.20, 0.15, and 0.10, respectively [41]. On the other hand, if we take $r = 0.1$, which is the correlation coefficient between S and the LZ complexity of Bob's bit strings for $N \leqslant 20\,000$, then the required sample size is 782 for the test with significance level $\alpha = 0.05$ (two-tailored) and power $1 - \beta = 0.8$.

IV. CONCLUSION

In this paper, we presented a no-go theorems (and its generalization) showing the contradiction between the existence of the CPRNGs and that of the efficiently computable randomness measures distinguishing the outcomes from the PRNGs and those from ECS systems. Given the success of modern cryptography which hinges on the presumed existence of the CPRNGs, it seems difficult to deny the assumption in practice, despite the fact that by definition PRNGs (including CPRNGs) are less random than ECS systems (see Sec. II A 3). Accordingly, we could say that there are no efficiently computable randomness measures capable of distinguishing the outcomes of PRNGs from those of the ECS systems. It follows that if the QRNG under consideration is ECS then these exists no random measures to distinguish the QRNG from a properly chosen PRNG.

In addition, we gave two comparative analyses between the PRNGs and the actual data on the quantum coin tosses and the Innsbruck experiment. In both analyses, we found that the average values of the algorithmic complexity measures for

TABLE II. Correlation analysis of the bit strings with the stratification of N. The Pearson correlation coefficient r between the nonlocality and randomness measures is calculated.

Stratification	Bit sequence	Randomness measure	r	p value	95% coincidence interval
$N < 20\,000$ (10 samples)	Alice	LZ	0.450	0.192	[−0.251, 0.841]
		Normalized LZ	0.405	0.246	[−0.302, 0.824]
		Borel normality	−0.313	0.379	[−0.787, 0.394]
	Bob	LZ	0.092	0.801	[−0.571, 0.682]
		Normalized LZ	0.115	0.751	[−0.555, 0.694]
		Borel normality	−0.080	0.825	[−0.676, 0.579]
	Mixed	LZ	0.448	0.194	[−0.253, 0.841]
		Normalized LZ	0.325	0.360	[−0.383, 0.792]
		Borel normality	−0.222	0.538	[−0.747, 0.474]
$20\,000 \leqslant N < 40\,000$ (7 samples)	Alice	LZ	0.395	0.381	[−0.510, 0.885]
		Normalized LZ	0.437	0.327	[−0.471, 0.895]
		Borel normality	−0.054	0.909	[−0.775, 0.729]
	Bob	LZ	0.047	0.921	[−0.732, 0.773]
		Normalized LZ	−0.014	0.977	[−0.759, 0.747]
		Borel normality	−0.178	0.703	[−0.821, 0.664]
	Mixed	LZ	−0.240	0.604	[−0.841, 0.626]
		Normalized LZ	−0.515	0.237	[−0.914, 0.389]
		Borel normality	−0.236	0.610	[−0.840, 0.629]
$N \geqslant 40\,000$ (3 samples)	Alice	LZ	−0.293	0.811	[−1, 1]
		Normalized LZ	−0.572	0.612	[−1, 1]
		Borel normality	0.075	0.952	[−1, 1]
	Bob	LZ	0.845	0.359	[−1, 1]
		Normalized LZ	0.879	0.316	[−1, 1]
		Borel normality	−0.975	0.144	[−1, 1]
	Mixed	LZ	0.684	0.520	[−1, 1]
		Normalized LZ	0.665	0.537	[−1, 1]
		Borel normality	−0.601	0.589	[−1, 1]

the PRNGs are indistinguishable from those for the QRNGs. This is an assuring result and is consistent with our theorem. In the meantime, we reconsidered the former analyses [24] which erroneously reported an anticorrelation between the randomness and nonlocality in the Innsbruck experiment. In contrast, we found that the randomness measures show no such correlation between locality and randomness in general. Instead, a clear correlation is found between the randomness measures and (joint) binary entropy, suggesting the existence of possible systematic errors in Alice's apparatus.

The fact that our theorem can be applied only to ECS quantum systems indicates that for QRNG the property of ECS may be an important characteristic when it comes to the study of the nature of randomness. Since this connection has been largely unnoticed in the existing literature [21–25], it would be of interest to find quantum systems which are not ECS and examine if their outcomes exhibit any signature different from PRNG.

ACKNOWLEDGMENTS

We thank G. Weihs for providing the data set of the Innsbruck experiment. This work was supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant No. JPMXS0120319794, and by JSPS KAKENHI Grants No. 20H01906 and No. JP22K13970.

APPENDIX A: NUMERICAL FEATURES OF THE MEASURES IN THE LONGDIST* SAMPLES

Table III shows the statistics of the longdist data sets in the Innsbruck experiment. The number of the pairs of the coincident outcomes N obtained in our analyses is almost identical to that obtained in the original analyses by Weihs [31] with a difference of about an order of 10, except the sample longdist4. In contrast, N given in [24] differs from the original analyses by Weihs about $O(10^3)$.

The exceptional behavior of the sample longdist4 is presumably due to a difference between the stored and actual threshold values. Indeed, in the original analyses [31], the stored threshold value and N for longdist4 are completely identical to those for longdist3, which is unlikely. Taking this possibility into account, we regarded longdist4 as an outlier and excluded it from our analyses.

APPENDIX B: PROOF OF THEOREM 2

In this Appendix we prove Theorem 2.

We begin by giving an overview of the proof. The basic idea here is the same as in the case of the biased bit generator in Sec. II D 1 (see Fig. 8).

As we assumed that system A is ECS, there exists a probabilistic algorithm A' which simulates A with arbitrary accuracy. Let $\bar{A}(1^n)$ be the algorithm obtained by repeating

TABLE III. Numerics of the measures.

			LZ			Normalized LZ			Borel normality		
File name	N	S	Alice	Bob	Mixed	Alice	Bob	Mixed	Alice	Bob	Mixed
longdist0	17 374	2.518	1.531	1.535	1.503	0.973	0.976	0.981	0.595	0.182	0.424
longdist1	17 640	2.629	1.531	1.534	1.499	0.974	0.975	0.978	0.462	0.166	0.331
longdist2	27 708	1.978	1.507	1.511	1.482	0.979	0.982	0.983	0.483	0.207	0.398
longdist3	27 620	2.660	1.510	1.511	1.480	0.981	0.981	0.982	0.499	0.221	0.364
longdist4	1112	1.810	0.819	1.665	1.380	0.464	0.943	0.812	7.158	1.110	3.690
longdist5	27 420	2.622	1.508	1.510	1.479	0.980	0.981	0.981	0.517	0.203	0.386
longdist10	28 248	2.283	1.503	1.507	1.477	0.977	0.980	0.981	0.641	0.370	0.516
longdist11	28 818	2.405	1.507	1.508	1.477	0.980	0.981	0.981	0.649	0.278	0.515
longdist12	28 718	2.393	1.503	1.508	1.477	0.978	0.981	0.982	0.636	0.254	0.521
longdist13	28 513	2.357	1.502	1.506	1.477	0.976	0.979	0.981	0.784	0.335	0.559
longdist20	43 315	2.058	1.491	1.489	1.463	0.977	0.976	0.978	0.166	0.341	0.272
longdist22	42 797	2.186	1.494	1.490	1.464	0.979	0.976	0.979	0.071	0.342	0.223
longdist23	42 778	2.632	1.491	1.491	1.464	0.976	0.976	0.979	0.143	0.265	0.228
longdist30	15 306	2.104	1.524	1.539	1.500	0.968	0.977	0.977	1.095	0.311	0.745
longdist31	15 122	2.637	1.517	1.542	1.500	0.963	0.979	0.977	1.168	0.283	0.756
longdist32	14 294	2.703	1.521	1.538	1.499	0.964	0.976	0.975	1.184	0.562	0.874
longdist33	15 113	2.055	1.519	1.541	1.498	0.964	0.978	0.975	1.170	0.404	0.794
longdist34	14 824	1.949	1.520	1.536	1.499	0.964	0.974	0.976	1.133	0.361	0.764
longdist35	14 573	2.728	1.523	1.542	1.500	0.967	0.979	0.976	1.198	0.338	0.745
longdist36	14 571	2.720	1.527	1.541	1.502	0.969	0.978	0.977	1.020	0.378	0.743
longdist37	14 673	1.953	1.518	1.540	1.498	0.963	0.978	0.975	1.241	0.366	0.743

$A'(1^n)$ n times. Since $\bar{A}(1^n)$ is also a probabilistic algorithm, it can be described as a deterministic algorithm having a TRNG $U^{l_G(n)}$ as an auxiliary input which $\bar{A}(1^n)$ uses for its internal coin tossing, with $l_G(n)$ being a polynomial of n. By replacing the auxiliary input $U^{l_G(n)}$ by a CPRNG $G(U^k)$ $(\in \{0,1\}^{l_G(k)})$, we obtain the desired algorithm $B(U^k)$.

To see that $B(U^k)$ thus obtained is indeed indistinguishable from the original physical system $A^{l(k)}$, first note that $\bar{A}(1^{l(k)})$ simulates the output of $A^{l(k)}$ with arbitrary accuracy, due to the ECS property of $\bar{A}(1^n)$. Also note that, by definition of the CPRNG, TRNG $U^{l_G(k)}$, and CPRNG $G(U^k)$ are indistinguishable from each other, and thus their outputs followed by an application of an identical algorithm $\bar{A}(1^{l(k)})$ are also indistinguishable; namely, the outputs of $\bar{A}(1^{l(k)})$ and $B(U^k)$ are indistinguishable. As a result, the outputs of $A^{l(k)}$ and $B(U^k)$ are indistinguishable.

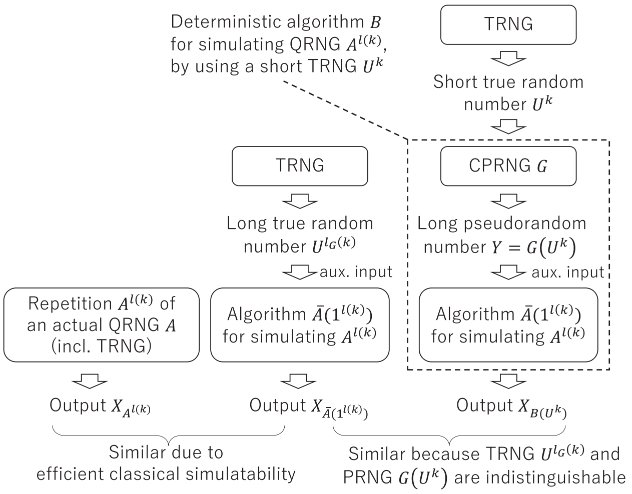
Deterministic algorithm B for simulating QRNG $A^{l(k)}$, by using a short TRNG U^k
TRNG
Short true random number U^k
TRNG
CPRNG G
Long true random number $U^{l_G(k)}$
Long pseudorandom number $Y = G(U^k)$
aux. input
aux. input
Repetition $A^{l(k)}$ of an actual QRNG A (incl. TRNG)
Algorithm $\bar{A}(1^{l(k)})$ for simulating $A^{l(k)}$
Algorithm $\bar{A}(1^{l(k)})$ for simulating $A^{l(k)}$
Output $X_{A^{l(k)}}$
Output $X_{\bar{A}(1^{l(k)})}$
Output $X_{B(U^k)}$
Similar due to efficient classical simulatability
Similar because TRNG $U^{l_G(k)}$ and PRNG $G(U^k)$ are indistinguishable

FIG. 8. Overview of the proof of Theorem 2. Algorithm $\bar{A}(1^n)$ is the one in which the classical simulation algorithm $A'(1^n)$ of A is repeated n times. Here we set $n = l(k)$.

We will below elaborate this discussion.

1. Construction of algorithm *B*

Following Definition 5, fix an efficient classical simulation of A, and denote it by A'.

Let $\bar{A}(1^n)$ be the algorithm obtained by repeating $A'(1^n)$ n times. Since $\bar{A}$ is a polynomial-time algorithm, the running time of $\bar{A}(1^n)$ can be bounded from above by a polynomial $g(n)$. Without loss of generality, we may assume that $g(n)$ is monotonically increasing.

Recall that probabilistic algorithms in general can be described as a deterministic algorithm having a random tape as auxiliary inputs (see, e.g., Ref. [33], Sec. 1.3.2.2). In this description, one can regard $\bar{A}(1^n)$ as a deterministic algorithm having TRNG $U^{l_G(n)}$ as an auxiliary input (to be used for its internal coin tossing), and halts within time $g(n)$.

We define algorithm B by replacing $\bar{A}(1^n)$'s auxiliary input TRNG $U^{l_G(n)}$ by a CPRNG $G(U^k)$. More precisely, choose a CPRNG $G : \{0,1\}^k \to \{0,1\}^{l_G(k)}$ with an expansion factor $l_G(k) = g(l(k))$ (the existence of such G is guranteed, e.g., by Ref. [33], Secs. 3.3.1 and 3.3.2, or Ref. [32], Theorem 8.7). Then define a deteriministic algorithm $B : \{0,1\}^* \to \mathcal{X}^*$ as

(1) On input a random seed $s \in \{0,1\}^k$, B inputs s to G and obtains the output $r \in \{0,1\}^{l_G(k)}$.

(2) B starts algorithm $\bar{A}$ with setting $1^{l(k)}$ as the input and s as the auxiliary input, and obtains the output $x = (x_1, \ldots, x_{l(k)}) \in \mathcal{X}^{l(k)}$. B outputs x.

2. Indistinguishability of $A^{l(k)}$ and $\bar{A}(1^{l(k)})$

In the above setting, the variational distance between the outputs of A^n and $\bar{A}(1^n)$ can be bounded as

$$\begin{aligned}
&\|P(X_{A^n})-P(X_{\bar{A}(1^n)})\|_1\\
&\quad\leqslant \|P(X_{A^n}^n)-P(X_AX_{\bar{A}(1^{n-1})})\|_1\\
&\qquad+\|P(X_AX_{\bar{A}(1^{n-1})})-P(X_{\bar{A}(1^n)}^n)\|_1\\
&\quad= \|P(X_A)\|\,\|P(X_{A_{n-1}})-P(X_{\bar{A}(1^{n-1})})\|_1\\
&\qquad+\|P(X_A)-P(X_A')\|_1\|P(X_{\bar{A}(1^{n-1})})\|_1\\
&\quad\leqslant \|P(X_{A_{n-1}})-P(X_{\bar{A}(1^{n-1})})\|_1+2^{-n}\\
&\quad\leqslant\cdots\leqslant n2^{-n}\leqslant \mathrm{negl}(n),
\end{aligned}\tag{B1}$$

where $\|P(X)\|_1$ denotes the trace norm of the probability distribution of X, $\|P(X)\|_1:=\frac{1}{2}\sum_{x\in\mathcal{X}}P(X=x)$. Thus, by letting $n=l(k)$ (according to the setting of the previous subsection), we have

$$\|P(X_{A^{l(k)}})-P(X_{\bar{A}(1^{l(k)})})\|_1\leqslant \mathrm{negl}(l(k))\leqslant \mathrm{negl}(k).\tag{B2}$$

Then by using the monotonicity of the variational distance (see, e.g., [42], Theorem 9.2), we have

$$|P(D_I(I(X_{A^{l(k)}}))=1)-P(D_I(I(X_{\bar{A}(1^{l(k)})}))=1)|\leqslant \mathrm{negl}(k).\tag{B3}$$

3. Indistinguishability of $\bar{A}(1^{l(k)})$ and $B(U^k)$

Next we define a distinguisher algorithm $D:\{0,1\}^*\to\{0,1\}$ for CPRNG having an expansion factor $l_G(k)$ as

(1) On input $r\in\{0,1\}^*$, D finds $k\in\mathbb{N}$ satisfying $l_G(k)=|r|$. If such k is not found, D halts.

(2) D starts $\bar{A}$ with setting $1^{l(k)}$ as input and r as the auxiliary input, and obtains the output $x^{l(k)}$.

(3) D inputs $x^{l(k)}$ to $D_I(I(\dots))$ and obtains the output $b\in\{0,1\}$. D outputs b.

To see that this algorithm is indeed efficient, note that each step above can be executed within a polynomial time of k, and that k is always less than the input length $l_G(k)$ of D, since $l_G(k)$ is an expansion factor of G.

As we assume that G is a CPRNG, we have

$$|P(D(G(U^k))=1)-P(D(U^{l_G(k)})=1)|\leqslant \mathrm{negl}(k).\tag{B4}$$

Also, by definition, the situations where one inputs $U^{l_G(k)}$ to D, and where one input $1^{l(k)}$ to $\bar{A}$ are identical. Thus we have

$$P(D(U^{l_G(k)})=1)=P(D(I(X_{\bar{A}(1^{l(k)})}))=1).\tag{B5}$$

Similarly, since the situations where one inputs $G(U^k)$ to D, and where one inputs U^k to B are identical,

$$P(D(G(U_k))=1)=P(D(I(B(U_k)))=1).\tag{B6}$$

Then by combining relations (B4), (B5), and (B6), we have

$$|P(D(I(X_{\bar{A}(1^{l(k)})}))=1)-P(D(I((B(U_k)))=1)|\leqslant \mathrm{negl}(k).\tag{B7}$$

4. Triangle inequality

Finally, by applying the triangle inequality to inequalities (B3) and (B7), we obtain (7).

[1] S. J. Freedman and J. F. Clauser, Experimental test of local hidden-variable theories, Phys. Rev. Lett. **28**, 938 (1972).

[2] A. Aspect, J. Dalibard, and G. Roger, Experimental test of Bell's inequalities using time-varying analyzers, Phys. Rev. Lett. **49**, 1804 (1982).

[3] W. Tittel, J. Brendel, H. Zbinden, and N. Gisin, Violation of Bell inequalities by photons more than 10 km apart, Phys. Rev. Lett. **81**, 3563 (1998).

[4] G. Weihs, T. Jennewein, C. Simon, H. Weinfurter, and A. Zeilinger, Violation of Bell's inequality under strict Einstein locality conditions, Phys. Rev. Lett. **81**, 5039 (1998).

[5] M. A. Rowe, D. Kielpinski, V. Meyer, C. A. Sackett, W. M. Itano, C. Monroe, and D. J. Wineland, Experimental violation of a Bell's inequality with efficient detection, Nature (London) **409**, 791 (2001).

[6] H. Sakai, T. Saito, T. Ikeda, K. Itoh, T. Kawabata, H. Kuboki, Y. Maeda, N. Matsui, C. Rangacharyulu, M. Sasano *et al.*, Spin correlations of strongly interacting massive fermion pairs as a test of Bell's inequality, Phys. Rev. Lett. **97**, 150405 (2006).

[7] M. Ansmann, H. Wang, R. C. Bialczak, M. Hofheinz, E. Lucero, M. Neeley, A. D. O'Connell, D. Sank, M. Weides, J. Wenner *et al.*, Violation of Bell's inequality in Josephson phase qubits, Nature (London) **461**, 504 (2009).

[8] M. Giustina, A. Mech, S. Ramelow, B. Wittmann, J. Kofler, J. Beyer, A. Lita, B. Calkins, T. Gerrits, S. W. Nam *et al.*, Bell violation using entangled photons without the fair-sampling assumption, Nature (London) **497**, 227 (2013).

[9] M. Giustina, M. A. M. Versteegh, S. Wengerowsky, J. Handsteiner, A. Hochrainer, K. Phelan, F. Steinlechner, J. Kofler, J.-A. Larsson, C. Abellán *et al.*, Significant-loophole-free test of Bell's theorem with entangled photons, Phys. Rev. Lett. **115**, 250401 (2015).

[10] B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. L. Vermeulen, R. N. Schouten, C. Abellán *et al.*, Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres, Nature (London) **526**, 682 (2015).

[11] L. K. Shalm, E. Meyer-Scott, B. G. Christensen, P. Bierhorst, M. A. Wayne, M. J. Stevens, T. Gerrits, S. Glancy, D. R. Hamel, M. S. Allman *et al.*, Strong loophole-free test of local realism, Phys. Rev. Lett. **115**, 250402 (2015).

[12] J. P. Dehollain, S. Simmons, J. T. Muhonen, R. Kalra, A. Laucht, F. Hudson, K. M. Itoh, D. N. Jamieson, J. C. McCallum *et al.*, Bell's inequality violation with spins in silicon, Nat. Nanotechnol. **11**, 242 (2016).

[13] W. Rosenfeld, D. Burchardt, R. Garthoff, K. Redeker, N. Ortegel, M. Rau, and H. Weinfurter, Event-ready Bell test using entangled atoms simultaneously closing detection and locality loopholes, Phys. Rev. Lett. **119**, 010402 (2017).

[14] D. Rauch, J. Handsteiner, A. Hochrainer, J. Gallicchio, A. S. Friedman, C. Leung, B. Liu, L. Bulla, S. Ecker, F. Steinlechner *et al.*, Cosmic Bell test using random measurement settings from high-redshift quasars, Phys. Rev. Lett. **121**, 080403 (2018).

[15] J. S. Bell, On the Einstein Podolsky Rosen paradox, Phys. Phys. Fiz. **1**, 195 (1964).

[16] D. Bohm, A suggested interpretation of the quantum theory in terms of "hidden" variables. I, Phys. Rev. **85**, 166 (1952).

[17] D. Bohm, A suggested interpretation of the quantum theory in terms of "hidden" variables. II, Phys. Rev. **85**, 180 (1952).

[18] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani *et al.*, Advances in quantum cryptography, Adv. Opt. Photonics **12**, 1012 (2020).

[19] M. Herrero-Collantes and J. C. Garcia-Escartin, Quantum random number generators, Rev. Mod. Phys. **89**, 015004 (2017).

[20] N. Gisin, *Quantum Chance: Nonlocality, Teleportation and Other Quantum Marvels* (Copernicus, Göttingen, USA, 2014).

[21] C. S. Calude, M. J. Dinneen, M. Dumitrescu, and K. Svozil, Experimental evidence of quantum randomness incomputability, Phys. Rev. A **82**, 022102 (2010).

[22] A. Solis, A. M. A. Martinez, R. R. Alarcon, H. C. Ramirez, A. B. U'Ren, and J. G. Hirsch, How random are random numbers generated using photons? Phys. Scr. **90**, 074034 (2015).

[23] A. C. Martinez, A. Solis, R. Diaz Hernandez Rojas, A. B. U'Ren, J. G. Hirsch, and I. Perez Castillo, Advanced statistical testing of quantum random number generators, Entropy **20**, 886 (2018).

[24] M. G. Kovalsky, A. A. Hnilo, and M. B. Agüero, Kolmogorov complexity of sequences of random numbers generated in Bell's experiments, Phys. Rev. A **98**, 042131 (2018).

[25] A. A. Abbott, C. S. Calude, M. J. Dinneen, and N. Huang, Experimentally probing the algorithmic randomness and incomputability of quantum randomness, Phys. Scr. **94**, 045103 (2019).

[26] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed., Wiley Series in Telecommunications and Signal Processing (Wiley-Interscience, 2006).

[27] A. Lempel and J. Ziv, On the complexity of finite sequences, IEEE Trans. Inf. Theory **22**, 75 (1976).

[28] F. Kaspar and H. G. Schuster, Easily calculable measure for the complexity of spatiotemporal patterns, Phys. Rev. A **36**, 842 (1987).

[29] E. Estevez-Rams, R. Lora Serrano, B. Aragón Fernández, and I. Brito Reyes, On the non-randomness of maximum Lempel Ziv complexity sequences of finite size, Chaos **23**, 023118 (2013).

[30] C. S. Calude, *Information and Randomness: An Algorithmic Perspective* (Springer, Berlin, 2002).

[31] G. Weihs, T. Jennewein, C. Simon, H. Weinfurter, and A. Zeilinger, Violation of Bell's inequality under strict Einstein locality conditions, (1998), doi:10.5281/zenodo.7185335.

[32] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed. (Chapman and Hall/CRC Press, Boca Raton, 2020).

[33] O. Goldreich, *Foundations of Cryptography* (Cambridge University Press, Cambridge, 2001), Vol. 1.

[34] R. Jozsa and N. Linden, On the role of entanglement in quantum-computational speed-up, Proc. R. Soc. London A **459**, 2011 (2003).

[35] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed. (Course Technology, Boston, 2013).

[36] Y. Nir and A. Langley, ChaCha20 and Poly1305 for IETF Protocols, Internet Research Task Force, https://www.rfc-editor.org/rfc/rfc8439 (2018).

[37] L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, S. Leigh, M. Levenson, M. Vangel, N. Heckert, and D. Banks, *A statistical test suite for random and pseudorandom number generators for cryptographic applications* (SP 800-22 Rev-1a, National Institute of Standards and Technology, Gaithersburg, 2010).

[38] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, Proposed experiment to test local hidden-variable theories, Phys. Rev. Lett. **23**, 880 (1969).

[39] B. S. Cirel'son, Quantum generalizations of Bell's inequality, Lett. Math. Phys. **4**, 93 (1980).

[40] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences* (Lawrence Erlbaum Associates, Hillsdale, 1988).

[41] M. Moinester and R. Gottfried, Sample size estimation for correlations with pre-specified confidence interval, Quantitative Methods Psychol. **10**, 124 (2014).

[42] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2010).